A Review of the Past, Present, and Future of Secure Data Deduplication

Tamanda Mgwalima
School of Information Science and Technology Department
Harare Institute of Technology, Zimbabwe

Arthur Ndlovu
School of Information Science and Technology Department
Harare Institute of Technology, Zimbabwe

***Abstract*—In today's digital world, many enterprises, organizations and individuals have chosen to outsource their data to cloud storage providers to reduce the burden of maintaining enormous amounts of data. Storage optimization techniques have become an essential requirement in cloud storage and many cloud storage providers perform de-duplication which avoids storing duplicate data copies from multiple users. Cloud subscribers do not rely on service providers for the security of their data. To ensure data confidentiality, data is first encrypted by cloud subscribers before being outsourced to the cloud and one problem with that is, we cannot apply deduplication on encrypted data. Encryption of the same data using different keys (by different subscribers) will result in different ciphertexts that will not allow the Cloud Service Provider (CSP) to carry out deduplication. Performing deduplication over encrypted data securely in the cloud is a challenging task. Various secure deduplication methods to overcome this challenge have been researched and in this paper we review these state-of-the-art methods. Finally, we outline future research directions facing deduplication-based storage systems.**

Keywords*—Cloud Storage, Deduplication, Proof of Ownership, Convergent Encryption;

I. INTRODUCTION

According to NIST, cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.[1]

The cloud allows users to store enormous amounts of data, which can be retrieved as and when required. Google Drive [2], SugarSync [3], OpenDrive [4] and Amazon S3 [5] are a few examples of cloud storage offerings. In general, Cloud Service Providers (CSPs) store a single copy of identical data received from multiple sources to optimize space. However, CSPs cannot distinguish identical data when the clients upload the data in an encrypted form using different keys. Performing encryption is essential to ensure the confidentiality of data, at the same time, performing deduplication is essential for achieving optimized storage. Hence, deduplication and encryption need to work in hand to hand to ensure data confidentiality and optimized storage. Various techniques and approaches used for deduplication over encrypted data are studied in this paper.

II. LITERATURE REVIEW OF SECURE DATA DEDUPLICATION METHODS

Douceur et al [6] first provided a solution for deduplication over encrypted data aiming to achieve both data confidentiality and deduplication. In their scheme, convergent encryption was introduced to derive the encryption key from the hash of the plaintext. Two users with two identical plaintexts will then obtain two identical ciphertexts since the encryption key is the same; hence the cloud storage provider will be able to perform deduplication on such ciphertexts and keep one instance in storage. However, since convergent encryption is a deterministic encryption scheme, it suffers against brute-force dictionary attacks if the file comes from a predictable plaintext space. Since the proposal of convergent encryption, many other schemes for encrypted-data deduplication have been proposed in the literature. Convergent encryption is also susceptible to many attacks like "confirmation of a file attack" and "learn the remaining information attack". In confirmation of a file attack, anyone who owns the same file will have the potential to prove that another user also possesses the same file.

Bellare, Keelveedhi and Ristenpart [7] formalized Message-Locked Encryption (MLE), a cryptographic primitive where the key used for encryption and decryption is derived from the message itself. However, MLE scheme is vulnerable to brute-force attacks. Bellare et al. firstly proposed DupLESS [8] to resist the above-mentioned brute-force attacks. The scheme introduced a key server and implemented rate-limiting strategy to resist brute-force attacks.

V Chouhan, S Peddoju and R Buyya [9] propose a secure and reliable cloud storage framework to deduplicate the encrypted data and key (dualDup framework) that optimizes the storage by eliminating the duplicate encrypted data from multiple users by extending DupLESS concept, and securely distributes the data and key fragments to achieve the privacy and reliability using Erasure Coding scheme.

Halevi, Harnik, Pinkas, and Shulman-Peleg [10] addressed the security problem in client-side deduplication whereby an entire file is represented by a hash value, and an adversary who gets this hash value can claim ownership of the file by [10] proposing the concept of Proofs of ownership (PoW) based on Merkle tree. The cloud server computes the Merkle tree of the uploaded file and stores the root hash value of the Merkle tree, then the clients prove file ownership by computing the corresponding sibling paths based on the cloud server request. If the clients compute requested paths correctly, the cloud server considers that they are in the possession of the file. However, this scheme has a large computational overhead. Following [10], Pietro and Sorniotti [11] propose another efficient PoW scheme by choosing the projection of a file onto some randomly selected bit-positions as the file proof. They addressed a major security risk in existing proof of ownership schemes where an adversary in possession of a fraction of the original file is able to claim possession of the entire file.

Authors in [12] propose a PoW scheme based on bloom filter which is flexible and scalable. This scheme is more efficient at the client side than the approach in [10] and more efficient at the server side than the scheme in [11].

Li et.al [13] first addressed the problem of achieving efficient and reliable key management when implementing convergent encryption. They proposed a baseline approach in which a user encrypts a file with a convergent key and uses a unique independent master key for encrypting the convergent keys. The encrypted data is then outsourced to the cloud server for storage. This approach generates a huge number of keys with the increasing number of users and hence an enormous amount of key storage space. Another challenge of this baseline approach is that if the master key is compromised, the stored data cannot be recovered, hence the master key is a single point of failure. To overcome this challenge, they also proposed Dekey, a new construction that implements Ramp secret sharing scheme in which users do not need to manage any keys on their own but instead securely distribute the convergent key shares across multiple servers. Singh et al [14] also proposed an approach to eliminate this single point of failure by distributing the convergent keys into multiple random looking shares using the Chinese Remainder Theorem based secret sharing and sending the shares across multiple key management servers. The key can also be recovered when a threshold number of shares is obtained from the key management server by execution of POW protocols. In their scheme, encrypted data is distributed at multiple servers into random looking shares based on the Permutation ordered binary number system.

Puzio et al [15] proposed ClouDedup, which in addition to the basic storage provider, a metadata manager and an additional server are defined. The server adds an additional encryption layer to prevent well-known attacks against convergent encryption and thus protect the confidentiality of the data; on the other hand, the metadata manager is responsible of the key management task since block-level deduplication requires the memorization of a huge number of keys.

PoW schemes worked well when the file is in plaintext. However, the privacy of the clients' data may be vulnerable to honest-but-curious attacks. To deal with this issue, the clients tend to encrypt files before outsourcing them to the cloud, which makes the existing PoW schemes inapplicable any more. Chao Yang [16] first proposed a secure zero-knowledge based client side deduplication scheme over encrypted files. The scheme achieves a high detection probability of the clients' misbehavior. They introduced a proxy re-encryption based key distribution scheme which ensures that the server knows nothing about the encryption key even though it acts as a proxy to help distributing the file encryption key. It also enables the clients who have gained the ownership of a file to share the file with the encryption key generated without establishing secure channels among them. Clients' private key cannot be recovered by the server or clients' collusion attacks during the key distribution phase.

Authors in [17] propose a secure and efficient client-side encrypted deduplication scheme (CSED) built upon message locked encryption (MLE), where a dedicated key server is employed to assist clients in generating MLE keys without leaking any information about the data to be encrypted to the key server. They integrated a Bloom filter-based proof of ownership (PoW) mechanism into CSED to resist illegal content distribution attacks.

Most secure deduplication schemes presume that all files need equal security, however, Stanek and Kencl [18] proposes a scheme which provides security to the data on basis of their popularity. Data owned by many cloud subscribers is known as popular data and data owned by a few subscribers is called unpopular data. Semantic security is provided for unpopular files and convergent encryption is used for popular files. Authors in [19] proposed PerfectDedup, to counter the weaknesses in convergent encryption by taking into account the popularity of the data segments. PerfectDedup takes advantage of the properties of Perfect Hashing in order to assure block-level deduplication and data confidentiality simultaneously.

C Guo, X Jiang, K R Choo and Y Jie [20] proposed R-Dedup, a randomized, secure, client-side deduplication that does not rely on an external third-party or require assistance from other peer users. By sharing a random value used to generate an encryption key for users who hold the same copy of a file, R-Dedup can resist brute-force attacks from both malicious cloud servers and subscribers. Data Verification in R-Dedup ensures data integrity and provides user authentication for the cloud server. Less computation overhead occurs at the client side since the complex calculations are handled by the cloud server.

[21] proposed DedupDUM, a deduplication scheme with dynamic user management which updates dynamic group users in a secure way and restricts unauthorised cloud users from sensitive data owned by valid users. Their scheme supports user revocation and new cloud user joining by exploiting re-encryption techniques and does not require a fully trusted third party.

Geeta C M et al [22] proposed Secure deduplication and virtual auditing of data in the cloud (SDVADC). SDVADC supports secure deduplication of information and effective virtual auditing of the files during the download process. The approach lowers the burden of data owner to audit files by himself or a third party auditor. They proposed an algorithm for file uploading and virtual auditing that allows a user to encrypt a file using randomized convergent encryption and outsource the ciphertext to the distributed server in the cloud service provider premises. The metadata information of the file is sent to the Virtual Auditing Entity (VAE) that consists of the metadata information of all the files uploaded to the distributed server. The CSP accepts the file, and checks for duplication. If the file is original, the CSP saves the file in storage and if the file is a duplicate, the CSP runs PoW convention with the user. When the user proves that he is an authorized person, then the CSP provides a link for the file existing in storage. During file download, the cloud user transmits a file request query to the CSP. The CSP sends the requested file to the VAE. The virtual auditing framework performs auditing of this file and sends the file attached with the auditing report to the user that shows whether the file has been modified or not.

R Miguel, K Mi and M Aung [23] proposed HEDup, a scheme that performs deduplication on encrypted data, with the aid of a key server deployed at the cloud service provider premises. Subscribers obtains data-encryption key from the key server through some homomorphic searching operations.

[24] propose a cloud data deduplication scheme based on certificateless proxy reencryption. Certificateless cryptography is applied to solve the problem of key escrow and to avoid situations where a key generation centre impersonates a cloud

subscriber to decrypt ciphertext. In this scheme, a convergent key is used to encrypt the plaintext and the resulting ciphertext is encrypted again using certificateless proxy reencryption and stored in the cloud server together with the reencryption key. The reencryption key is used to share the encrypted data with other cloud subscribers authenticated by the POW scheme based on certificateless signature.

Authors in [25] propose an identity based proxy re-encryption scheme for cloud data deduplication. This scheme integrates cloud deduplication with access control unlike other solutions which cannot support flexible data access control and require data users to remain online. It efficiently distributes ciphertexts and revoke data owners who delete their data from the cloud without participation of the data owners unlike in certificateless proxy reencryption.

III. COMPARATIVE ANALYSIS

Client-side deduplication has been applied in most of the existing deduplication than server side deduplication. Table 1 gives the advantages and limitations of a few selected deduplication schemes and the methodologies behind the schemes.

Paper Title	Proposed Idea	Methodology	Advantages	Limitations
DupLESS : Server-Aided Encryption for Deduplicated Storage[8]	To resist brute force attack on Message Locked Encryption	The scheme combines has the ability to obtain message-derived keys with the help of a key server shared amongst a group of clients. The clients interact with the key server by a protocol for oblivious PRFs, ensuring that the key server can cryptographically mix in secret material to the per-message keys while learning nothing about files stored by clients.	It provides more security than convergent encryption	When there is significantly less deduplication across the corpus, DupLESS may introduce greater overhead.
ClouDedup : Secure Deduplication with Encrypted Data for Cloud Storage[15]	To prevent well known attacks against convergent encryption	Secret keys can be generated in a hardware-dependent way by the device itself Server encryption is applied on top of convergent encryption performed by user	It prevents curious cloud storage providers from inferring the original content of stored data by observing access patterns or accessing metadata. It gains in terms of storage space are not affected by the overhead of metadata management, which is minimal.	If the third party server is comprised, it may lead a Man-in-the-Middle attack to the users.
Secure Data Deduplication in Cloud Storage Services Doctoral Thesis[18]	Data can be differentiated based on popularity to determine the level of privacy offered.	Semantic security is provided for all the unpopular data whereas for popular data the security is slightly weaker convergent encryption is provided for popular data	The users no longer need to manually classify sensitive files. The transition between unpopular and popular state is automatic and does not require active user participation	Lower deduplication ratio. High computation cost
CSED : Client-Side encrypted deduplication scheme based on proofs of ownership for cloud storage [17]	To construct a secure and efficient scheme that resists brute-force attacks and illegal content distribution attacks, where the adversary can distribute data to other users via the cloud server.	Client-side encrypted deduplication scheme based on proofs of ownership (PoW) that resists brute-force attacks. A key server is employed to assist the clients in generating the MLE keys and adopt the rate-limiting strategy to prevent brute-force attacks. Bloom filter and hierarchical strategy on cloud storage to improve the efficiency.	Reduced storage and communication overhead. Secure against brute force attacks.	No integrity auditing.
HEDup : Secure Deduplication with Homomorphic Encryption[23]	Deduplication on encrypted data, with the aid of a key server deployed at the CSP premises. Client obtains the encryption key from the key server through some homomorphic searching operations. Data owners maintain exclusive control of their data and cloud providers has no access to any of it.	Allow deduplication on encrypted data with the aid of a key server deployed at cloud service. The subscriber encrypts data with data-encryption key obtained from key server via various key-management schemes, one of which uses homomorphic encryption. The key server deployed at cloud provider premises, it will not only deduplicate data from particular domain but also for the CSP's entire client base including public and different enterprise users	Data uploads and downloads using HEDup have minor storage and latency overhead. Data owners still maintain exclusive control of their data and data-encryption keys, i.e. CSP has no access to any of it -	Key server discussed in this approach may become a bottleneck when number of clients increase in case of large scale deployment, and a decentralized deployment of key server is supposed as a solution.
R-Dedup: Secure client-side deduplication for encrypted data without involving a	A randomized, secure, cross-user deduplication scheme that does not involve any third-party entity or require assistance	ElGamal Encryption technique Sharing a random value used to generate encryption key for users	provides user authentication and integrity of data	computation overhead on the client side

third-party entity[20]	from other users. In			
Zero knowledge based client side deduplication for encrypted files of secure cloud storage in smart cities[16]	A secure zero-knowledge based client side deduplication scheme over encrypted files is proposed.	Scheme enables a client to prove its file ownership via the original file without leaking any information to the server. Key distribution scheme is based on proxy re-encryption which realizes delegation of decryption rights.	Great detection probability of clients' misbehaviour	
Secure Deduplication with Efficient and Reliable Convergent Key Management[13]	To efficiently and reliably manage a huge number of convergent keys. Users do not need to manage any keys on their own but instead securely distribute the convergent key shares across multiple servers.	Scheme implements Ramp secret sharing scheme in which users do not need to manage any keys on their own but instead securely distribute the convergent key shares across multiple servers.	It incurs small encoding/decoding overhead com- pared to the network transmission overhead in the regular upload/download operations	No integrity auditing
An identity-based proxy re-encryption for data deduplication in cloud[25]	To design a scheme that can flexibly support ciphertext distribution even when the data owner is offline	The scheme saves only one copy of data in the cloud for the initial uploader and subsequent data owners who will have passed the proof of ownership will require conversion of this data into ciphertext that can be decrypted with their own private keys. (Proxy re-encryption).	The scheme successfully flexibly supports ciphertext distribution even when the data owner is offline	Slightly higher computation overhead of re-encryption
Authorized Client-Side Deduplication Using CP-ABE in Cloud Storage [26]	To allow only authorized users to access critical data. To provide control over access permissions in an encrypted deduplication storage	Ciphertext-Policy Attribute-Based Encryption (CPABE)	Less Authorization Server's burden and less storage overhead	time complexity
SecDep: A User-Aware Efficient Fine-Grained Secure Deduplication Scheme with Multi-Level Key Management [27]	To resist brute force attack and convergent key space overhead.	User-Aware Convergent Encryption (UACE) and Multi-Level Key management (MLK)	Time efficient and key-space-efficient	No integrity auditing and public verifiability

a. Table 1: Comparative Analysis of deduplication techniques

IV. CONCLUSION

Various secure deduplication techniques for providing in cloud storage have been discussed. Client-side has comparatively more benefits than server side Deduplication and hence it has been applied in most of the existing deduplication tools. This paper discussed the various secure data deduplication techniques, and outlined a comparative analysis of some of the different existing client-side deduplication schemes are done. Future enhancements might be designing a scheme capable of verifying integrity of data without downloading it from server, with reduced computation complexity and that supports insertion, deletion and updation operations, private verifiability, public verifiability and batch auditing and searchable encryption.

REFERENCES

[1] C. D. National Institute of Standards and Technology (NIST), "The NIST Definition of Cloud Computing," govinfo.gov, Jan. 2011, Accessed: Oct. 25, 2022. [Online]. Available: https%3A%2F%2Fwww.govinfo.gov%2Fapp%2Fdetails%2FGOVPUB-C13-74cdc274b1109a7e1ead7185dfec2ada

[2] "My Drive - Google Drive." https://drive.google.com/drive/my-drive (accessed Nov. 28, 2022).

[3] "Cloud File Sharing, File Sync & Online Backup From Any Device | SugarSync | SugarSync." https://www1.sugarsync.com/ (accessed Nov. 28, 2022).

[4] G. Xu, M. Lai, J. Li, L. Sun, and X. Shi, "Open Access A generic integrity verification algorithm of version files for cloud deduplication data storage," 2018.

[5] "Cloud Object Storage – Amazon S3 – Amazon Web Services." https://aws.amazon.com/s3/ (accessed Nov. 28, 2022).

[6] J. R. Douceur, A. Adya, W. J. Bolosky, D. Simon, and M. Theimer, "Reclaiming space from duplicate files in a serverless distributed file system," Proc. - Int. Conf. Distrib. Comput. Syst., pp. 617–624, 2002, doi: 10.1109/ICDCS.2002.1022312.

[7] M. Bellare, "Message-Locked Encryption and Secure Deduplication," pp. 1–29, 2013.

[8] M. Bellare, S. Keelveedhi, S. Diego, T. Ristenpart, W. Madison, and T. Ristenpart, "DupLESS : Server-Aided Encryption for Deduplicated Storage," 2013.

[9] V. Chouhan, S. K. Peddoju, and R. Buyya, "Journal of Information Security and Applications dualDup : A secure and reliable cloud storage framework to deduplicate the encrypted data and key," vol. 69, no. July, 2022.

[10] S. Halevi, D. Harnik, B. Pinkas, and A. Shulman-peleg, "Proofs of Ownership in Remote Storage Systems ∗," pp. 1–13, 2013.

[11] R. Di Pietro and A. Sorniotti, "Boosting efficiency and security in proof of ownership for deduplication," 2012.

[12] J. Blasco, A. Orfila, R. D. I. Pietro, and A. Sorniotti, "A Tunable Proof of Ownership Scheme for Deduplication Using Bloom Filters," pp. 1–18, 2014.

[13] J. Li, X. Chen, M. Li, J. Li, P. P. C. Lee, and W. Lou, "Secure Deduplication with Efficient and Reliable Convergent Key Management," vol. 25, no. 6, pp. 1615–1625, 2014.

[14] P. Singh, N. Agarwal, and B. Raman, "Secure data deduplication using secret sharing schemes over cloud," Futur. Gener. Comput. Syst., vol. 88, pp. 156–167, 2018, doi: 10.1016/j.future.2018.04.097.

[15] P. Puzio and S. Loureiro, "ClouDedup : Secure Deduplication with Encrypted Data for Cloud Storage".

[16] C. Yang, M. Zhang, Q. Jiang, J. Zhang, and D. Li, "Zero knowledge based client side deduplication for encrypted files of secure cloud storage in smart cities ☆," Pervasive Mob. Comput., vol. 41, pp. 243–258, 2017, doi: 10.1016/j.pmcj.2017.03.014.

[17] S. Li, C. Xu, and Y. Zhang, "Journal of Information Security and Applications CSED : Client-Side encrypted deduplication scheme based on proofs of ownership for cloud storage," vol. 46, pp. 250–258, 2019, doi: 10.1016/j.jisa.2019.03.015.

[18] D. Thesis and I. Technology, "Secure Data Deduplication in Cloud Storage Services Doctoral Thesis," no. March, 2018.

[19] P. Puzio, R. Molva, M. Onen, and S. Loureiro, "PerfectDedup : Secure Data Deduplication," vol. 2020, no. 644412, pp. 1–18.

[20] C. Guo, X. Jiang, K. R. Choo, and Y. Jie, "Journal of Network and Computer Applications R-Dedup : Secure client-side deduplication for encrypted data without involving a third-party entity," J. Netw. Comput. Appl., vol. 162, no. April, p. 102664, 2020, doi: 10.1016/j.jnca.2020.102664.

[21] H. Yuan, X. Chen, T. Jiang, X. Zhang, and Y. Xiang, "DedupDUM: Secure and Scalable Data Deduplication With Dynamic User Management," Inf. Sci. (Ny)., vol. 456, 2018, doi: 10.1016/j.ins.2018.05.024.

[22] C. M. Geeta, S. R. R. G, and K. R. Venugopal, "ScienceDirect SDVADC : SDVADC : Secure Secure Deduplication Deduplication and and Virtual Virtual Auditing Auditing of of Data Data in in Cloud," Procedia Comput. Sci., vol. 171, no. 2019, pp. 2225–2234, 2020, doi: 10.1016/j.procs.2020.04.240.

[23] R. Miguel, K. Mi, and M. Aung, "HEDup : Secure Deduplication with Homomorphic Encryption," pp. 215–223, 2020.

[24] X. Zheng, Y. Zhou, Y. Ye, and F. Li, "A cloud data deduplication scheme based on certificateless proxy re-encryption," J. Syst. Arch., vol. 102, 2020.

[25] G. Kan, C. Jin, H. Zhu, Y. Xu, and N. Liu, "An identity-based proxy re-encryption for data deduplication in cloud," J. Syst. Archit., vol. 121, no. October, p. 102332, 2021, doi: 10.1016/j.sysarc.2021.102332.

[26] T. Youn, N. Jho, K. H. Rhee, and S. U. Shin, "Authorized Client-Side Deduplication Using CP-ABE in Cloud Storage," vol. 2019, 2019.

[27] Y. Zhou et al., "SecDep: A User-Aware Efficient Fine-Grained Secure Deduplication Scheme with Multi-Level Key Management," 2015. doi: 10.1109/MSST.2015.7208297.